

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

APLIKACE PRO GEOLOKAČNÍ DATABÁZE

APPLICATION FOR GEOLOCATION DATABASES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

František Farkaš

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Dan Komosný, Ph.D.

BRNO 2017



Bakalářská práce

bakalářský studijní obor **Teleinformatika**

Ústav telekomunikací

Student: František Farkaš

ID: 164835

Ročník: 3

Akademický rok: 2016/17

NÁZEV TÉMATU:

Aplikace pro geolokační databáze

POKyny PRO VYPRACOVÁNÍ:

Vytvořte aplikaci, která bude provádět odhad geografické polohy pro zadané IP adresy. K tomu účelu využijte veřejně dostupné služby zvolených IP geolokačních databází. Sestavte seznam IP adres, u kterých je známa jejich poloha minimálně na úroveň města. Pro tento seznam vyhodnoťte přesnost odhadu v kilometrech. Správné a odhadnuté polohy vykreslete do mapy na webové stránce. Aplikaci sestavte v programovacím jazyce Python.

DOPORUČENÁ LITERATURA:

[1] PUŽMANOVÁ, R. TCP/IP v kostce. 2. vyd. Kopp, 2009. 620 s. ISBN: 978-80-7232-388-3.

[2] POESE, I., UHLIG, S., KAAFAR, M., DONNET, B., GUEYE, B. IP Geolocation Databases: Unreliable? ACM SIGCOMM Computer Communication Review, 2011, roč. 41, č. 2, s. 53-56. ISSN: 0146-4833.

Termín zadání: 1.2.2017

Termín odevzdání: 8.6.2017

Vedoucí práce: doc. Ing. Dan Komosný, Ph.D.

Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Témou tejto práce je testovanie a analýza geolokačných databáz s následným porovnaním medzi komerčnými a nekomerčnými databázami. Súčasťou práce je vytvorenie aplikácie využitej na testovanie geolokačných databáz a následné grafické zobrazenie výsledkov. V prvej časti práce je popísaná geolokácia, metódy získavania geolokačných dát a jednotlivé testované databázy. V druhej kapitole práce je bližšie vysvetlená implementácia aplikácie a použité technológie. Posledná kapitola sa venuje analýze výsledkov testovania a porovnávaniu databáz.

KĽÚČOVÉ SLOVÁ

Geolokácia, Databáza, IP adresa, Python

ABSTRACT

The topic of this thesis is testing and analysis of geolocation databases and comparison between commercial databases and noncommercial ones. Practical part is dedicated to implementation of application which is testing the databases and gathers the output data for analysis. First chapter of the thesis describes geolocation, use of geolocation and methods of getting geolocation data. Second chapter deals with implementation details of application and technologies used for implementation. The last chapter is dedicated to analysis and comparison of tested databases.

KEYWORDS

Geolocation, Database, IP address, Python

FARKAŠ, František *Aplikace pro geolokační databáze* : bakalárska práca. BRNO: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2017. 50 s. Vedúci práce bol prof. Ing. Dan Komosný, Ph.D.

PREHLÁSENIE

Prehlasujem, že som svoju bakalársku prácu na tému „Aplikace pro geolokační databáze“ vypracoval(a) samostatne pod vedením vedúceho bakalárskej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil(a) autorské práva tretích osôb, najmä som nezasiahol(-la) nedovoleným spôsobom do cudzích autorských práv osobnostných a/nebo majetkových a som si plne vedomý(-á) následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., o právu autorskom, o právach súvisiacich s právom autorským a o zmeně niektorých zákonov (autorský zákon), vo znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhé, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

BRNO

.....
podpis autora

POĎAKOVANIE

Na tomto mieste by som chcel rád poďakovať všetkým ľuďom bez ktorých by táto práca nemohla vzniknúť. Menovite chcem poďakovať vedúcemu bakalárskej práce pánovi doc. Ing. Danovi Komosnému, Ph.D za odborné vedenie, konzultácie a podnetné návrhy k práci. Tiež sa chcem poďakovať vedeniu Technickej univerzity v Brne za nadobudnuté vedomosti počas doterajšieho štúdia. Rád by som chcel poďakovať svojim rodičom a blízkym za podporu a pomoc bez ktorej by toto štúdium nebolo možné. Taktiež by som ešte chcel poďakovať slečne Kataríne Lemerovej za pomoc pri korekcii textu a podporu počas písania prace

BRNO

.....
podpis autora

POĎAKOVANIE

Výzkum popsaný v tejto bakalárskej práci bol realizovaný v laboratóriách podporených projektom SIX; registračné číslo CZ.1.05/2.1.00/03.0072, operačný program Výzkum a vývoj pro inovace.

BRNO

.....
podpis autora

OBSAH

Úvod	13
1 Geolokačné služby	14
1.1 Využitie geolokácie	15
1.2 Presnosť IP geolokácie	16
1.3 Pasívna metóda	16
1.4 Aktívna metóda	17
1.4.1 GeoPing	17
1.4.2 Geolokácia založená na ohraničení	18
2 Databázy	19
2.1 Voľne dostupné databázy	19
2.1.1 freegeoip.net	19
2.1.2 MaxMind	19
2.1.3 IP2Location	20
2.1.4 HostIP	20
2.1.5 DB-IP(IP to City)	20
2.2 Komerčné databázy	21
2.2.1 DB-IP(IP to Location)	21
2.2.2 Geobytes	21
2.2.3 IP Info	21
2.2.4 MaxMind	21
2.2.5 Skyhook	22
2.2.6 Eurek API	22
3 Návrh aplikácie	23
3.1 Parsovanie vstupných dát	24
3.1.1 Príklad vstupných dát	25
3.1.2 Získavanie geolokačných dát	28
3.1.3 Pravidlá pre úpravu dát	31
3.2 Implementácia spracovania dát	32
4 Analýza výstupných dát	37
4.1 Výsledky testovania množiny DNS serverov	37
4.1.1 Zhoda na úrovni krajiny	37
4.1.2 Zhoda na úrovni regiónu	38
4.1.3 Zhoda na úrovni mesta	39
4.2 Výsledky testovania množiny PlanetLab.	40

4.2.1	Zhoda na úrovni krajiny	40
4.2.2	Zhoda na úrovni regiónu	41
4.2.3	Zhoda na úrovni mesta	41
4.3	Porovnanie dosiahnutých výsledkov	42
5	Záver	45
	Literatúra	46
	Zoznam symbolov, veličín a skratiek	48
	Zoznam príloh	49
A	Obsah priloženého CD	50

ZOZNAM OBRÁZKOV

1.1	Príklad metódy CBG prevzaté z [14]	18
3.1	Diagram aplikácie.	23
3.2	Rozloženie testovaných DNS serverov po kontinentoch.	26
3.3	Rozloženie testovaných PlanetLab serverov po kontinentoch	27
3.4	Diagram modulu <code>fileprocessing.py</code> .	33
3.5	Príklad mapy generovanej skriptom pre jeden záznam	35
3.6	Príklad mapy generovanej skriptom pre jeden záznam.	36
4.1	Výsledky zhody na úrovni krajiny pre DNS.	38
4.2	Výsledky zhody na úrovni regiónu pre DNS.	39
4.3	Výsledky zhody na úrovni mesta pre DNS.	39
4.4	Výsledky zhody na úrovni krajiny pre PlanetLab.	40
4.5	Výsledky zhody na úrovni regiónu pre PlanetLab.	41
4.6	Výsledky zhody na úrovni mesta pre PlanetLab.	42
4.7	Kumulatívna distribučná funkcia pre množinu DNS.	43
4.8	Kumulatívna distribučná funkcia pre množinu PlanetLab.	44

ZOZNAM TABULIEK

3.1	Príklad vstupných dát pre množinu DNS.	27
4.1	Výsledky testovanie pre množinu DNS.	37
4.2	Výsledky testovanie pre množinu PlanetLab.	40
4.3	Kvartily chýb jednotlivých databáz pre množinu DNS serverov.	43
4.4	Kvartily chýb jednotlivých databáz pre množinu PlanetLab.	44

ZOZNAM VÝPISOV KÓDU

3.1	Príklad štruktúry vstupných dát	25
3.2	Definícia funkcie pre získavanie dát z geolokačných databáz.	28
3.3	Príklad výpočtu chyby.	29
3.4	Príklad zápisu do súboru.	30
3.5	Príklad vytvárania objektu request.	31
3.6	Obsah objektu request.	31
3.7	Príklad spustenia skriptu.	34

ÚVOD

Cielom tejto práce je vytvoriť aplikáciu na získavanie informácií o geografickej polohe sieťového zariadenia podľa jeho IP adresy (verzie IPv4) pomocou platených ale aj voľne dostupných geolokačných databáz. V tejto práci je ako prvé rozobratá geolokácia a princípy vyhľadávania pomocou IP adresy, porovnanie jednotlivých databáz a napokon návrh aplikácie na získavanie a analýzu získaných dát. V prvej kapitole sú okrem geolokácie ako celku aj bližšie popísané pasívne a aktívne metódy získavania geografickej polohy. Druhá kapitola sa bližšie sústreďuje na popis testovaných databáz a ich porovnanie a krátke oboznámenie s API (Application Programming Interface – Programovateľné aplikačné rozhranie) týchto databáz. Kapitola 3 popisuje návrh tohto systému a detailný popis získavania dát z jednotlivých databáz a popis vstupných dát aplikácie. Ako posledné je v tejto práci analýza výstupných dát.

1 GEOLOKAČNÉ SLUŽBY

Geolokácia sa v dnešnej dobe teší veľkému využitiu v rôznorodých aplikáciach kde je potrebné využitie fyzickej polohy sieťového zariadenia užívateľa. Využitie tohto poznatku je väčšinou využívané na cielené komerčne účely (reklama lokálnych obchodov, služieb a pod.). Geolokačné aplikácie majú za úlohu dve veci: hlásiť polohu užívateľa (tá môže byť získaná pomocou zariadení používajúcich GPS ako napríklad smartfóny, alebo pomocou počítača) a následne ju asociovať s geografickou polohou zariadenia. Zistenie tejto lokácie je náročnejšie než sa na prvý pohľad zdá, keďže celosvetové siete sú stavané na identifikáciu pripojeného sieťového zariadenia, ale nie na zistenie lokácie tohto zariadenia. Práve to viedlo rôzne súkromné spoločnosti a vyhľadávacích gigantov (napr. Google, Bing, Yahoo) k vytvoreniu systémov na analýzu lokácie zariadenia pomocou jeho IP adresy. Historicky bol ako prvý zdroj geolokačných informácií použitý DNS záznam sieťového zariadenia, ktorý sa ukázal byť veľmi nepresný, keďže z tohto záznamu nie je možné určiť exaktnú polohu[12]. Jednotlivé moderné metódy získavania geolokačných dát sú popísané v častiach 1.3 a 1.4.

IP adresa je numericky daná hodnota, ktorá je priradená každému zariadeniu, ktoré je účastníkom počítačovej siete. Na komunikáciu využíva protokol IP. Tento protokol je základným prenosovým mechanizmom, ktorý sa stará o prenos dát, a činí tak cez najnižšiu úroveň, vo všetkých sieťach využívajúcich protokol TCP/IP. Protokol IP je jednotnou nadstavbou rôznych prenosových technológií, ktoré zaisťujú fyzický prenos dát (Ethernet, FDDI). Protokol IP využíva tieto odlišné technológie k tomu, aby všade ponúkal jednotné prenosové služby rovnakej kvality.[1]

Na svete existuje približne 3.5 miliard smerovateľných IP adries[1]. Rôzne registre poskytujú verejne prístup k IP dátam, ale nešpecifikujú a negarantujú presnosť alokácií týchto adries. IP geolokačné dáta sú väčšinou presné na 8 až 16 metrový okruh [12] v Európe a Spojených štátoch. Presnosť závisí na procese získania IP dát od poskytovateľa, krajiny a typu siete.¹

Najzákladnejšia forma geolokácie spočíva v prechádzaní verejných registrov adries, agregovaní dát a následné uverejnenie či už voľne dostupné, alebo za finančný poplatok. Tento spôsob funguje pre rôzne spomínané komerčné služby. Pre presnejšie dešifrovanie geolokačných dát využívajú spoločnosti zaoberajúce sa touto problematikou sofistikované proprietárne platformy na analýzu týchto verejne prístupných adries. Tieto platformy využívajú softvérovú implementáciu dvoch metód lokalizácie, a to aktívnu a pasívnu.[13]

¹<https://www.nro.net/news/ipv4-free-pool-depleted>

1.1 Využitie geolokácie

Využitie IP geolokácie slúži najmä na komerčné účely ako napríklad:

- Cílené reklamy obchodníkov v lokálnej sfére.
- Banky zbierajú geolokacne dáta pri rozhodovaní o umiestnení bankomatov.
- Turistické spoločnosti za účelom zlepšiť svoje služby.
- Logistické spoločnosti pri analýze prepravy.

Okrem komerčných využití geolokakačných dát, sú tieto dáta využívané aj vládnyimi inštitúciami, napríklad na zisťovanie vzorov mestskej dopravy. Predpokladá sa, že tieto dáta sú takisto využívané aj bezpečnostnými zložkami za účelom štátnej bezpečnosti a vyhľadávanií hľadaných osôb. Tieto poznatky však nemožno povedať s istotou a jedná sa len o domnienky rôznych štúdií tejto problematiky, keďže žiadne využitie geolokačných dát bezpečnostnými zložkami nebolo oficiálne zverejnené.[3]

Veľké množstvo poskytovateľov geolokačných služieb sa spolieha na voľné dostupné registrované dáta, ktoré sú podľa rôznych štúdií v tomto smere nepresné.[12]. Nepresnosti tiež vznikajú aj na úrovni krajín, kde môžu byť chyby vyššie od 20 až 30%.

Tieto chyby vznikajú najmä z dôvodov ako napríklad:

- Voľná dostupnosť IP registrácie pomocou protokolu WHOIS pre poskytovateľov internetových služieb. WHOIS je protokol používaný na dotazovanie a odozvu serverov ktoré obsahujú informácie o registrovaných užívateľoch internetových služieb. Jedná sa o informácie ako doménový server, IP adresový blok alebo autonómny systém. Protokol je taktiež využívaný na širší rozsah informácií. WHOIS je definovaný v RFC 3912[15]².
- Väčšina poskytovateľov internetových služieb registruje celé alokované bloky IP adres na adresu svojej korporátnej adresy a nie na svojich koncových užívateľov, ktorým sú adresy pridelené.[11]

Alternatívne, moderné technologické riešenia využívajú čoraz častejšie takzvaný web spidering a iné metódy na trianguláciu geolokácie, rýchlosti pripojenia a iné charakteristiky asociované s IP adresami, kolektívne referovanému ako IP inteligencia, kde sa využíva vnútorná analýza siete s vonkajšou spätnou väzbou od užívateľa, za účelom vylepšenia a validácie odozvy na úrovni mesta resp. úrovni poštového smerovacieho čísla. Tento prístup umožňuje referencovanie lokácií koncových zariadení miesto lokácie sídla internetového poskytovateľa. Taktiež umožňuje vysokú presnosť na úrovni krajiny až 99% a úrovni mesta až 97%.[16]

²<https://tools.ietf.org/html/rfc3912>

1.2 Presnosť IP geolokácie

Existujú rôzne nie-IP geolokačné technológie, ktoré poskytujú informácie o lokácií v granulárnych kúskoch ako napríklad GPS ktorého návratová hodnota je zemepisná dĺžka a šírka. Lokačné dáta bývajú poskytnuté užívateľom pomocou HTTP (Hypertext Transfer Protocol – Protokol na výmenu hypertextových dokumentov) Cookies (Cookie v protokole HTTP označuje malé množstvo dát, ktoré posiela webový server prehliadaču), ktoré na koncových užívateľských zariadeniach ukladajú informácie o lokácií. Cookies sa ale čoraz viac považujú za invazívne voči užívateľovi.[14]

Na druhej strane pomocou IP geolokácie je možné zistiť lokáciu sieťového zariadenia do presnosti 5 až 8 kilometrovom rádiuse v celej internetovej sieti.

1.3 Pasívna metóda

Tento prístup k získavaniu dát využíva databázy poskytovateľov geolokačných služieb obsahujúcich záznamy IP adries, takzvané bloky. Tieto databázy poskytujú informácie o geografickej presnosti. Pasívne metódy geolokácie je možno rozlíšiť na tieto tri podmetódy podľa:

- IP adries.
- DNS záznamu.
- S využitím bezdrôtových sietí.

IP adresové bloky sú namapované na geografickú polohu. Pre toto mapovanie blokov sa využívajú smerovacie protokoly. Najčastejšie protokol BGP (Border Gate Protocol). Geolokačné databázy môžu byť buď komerčné alebo nekomerčné. Rozdiel medzi nimi je v licenčných podmienkach a často aj v presnosti a obmedzení istých funkcionalít. Tieto databázy získavajú svoje záznamy tak, že svoje záznamy adries porovnávajú s databázou organizácie IANA (Internet Assigned Numbers Authority), ktorá má na starosti pridelenie IP adries. Geolokačné databázy však trpia istými nedostatkami v mapovaní IP blokov a to kvôli tomu, že dochádza k odlišnostiam v miestach kde je obmedzenejšia sieťová infraštruktúra. Geolokácia na základe IP adresy sa považuje za najjednoduchšiu. Ďalsia metóda získavania polohy zariadenia je pomocou reverzného DNS záznamu sieťového zariadenia. DNS má hierarchickú štruktúru, čo umožňuje z tohto záznamu určiť krajinu, kde sa hľadané zariadenie nachádza.[14]

V dnešnej dobe je veľmi často zaužívané určovanie polohy pomocou bezdrôtového pripojenia. Pre určenie polohy pomocou tejto metódy je využívaný webový prehliadač, ktorý načítava informácie o okolitých prístupových bodoch ako MAC adresa a SSID siete. Tieto poznatky sa porovnávajú s databázou GoogleMaps. Databáza

si tieto informácie porovná s okolitými známymi bodmi, zmeria ich úroveň signálov a vráti približnú polohu zariadenia.[12]

Tieto databázové systémy sú založené na databázových systémoch ako napríklad rôzne implementácie SQL alebo NoSQL technológií. Vývojári týchto platforiem ponúkajú webové rozhranie, ktoré slúži na prístup k týmto databázovým záznamom. Toto rozhranie dovoľuje porovnať IP adresu vloženú užívateľom so záznamami databázy poskytovateľa.

1.4 Aktívna metóda

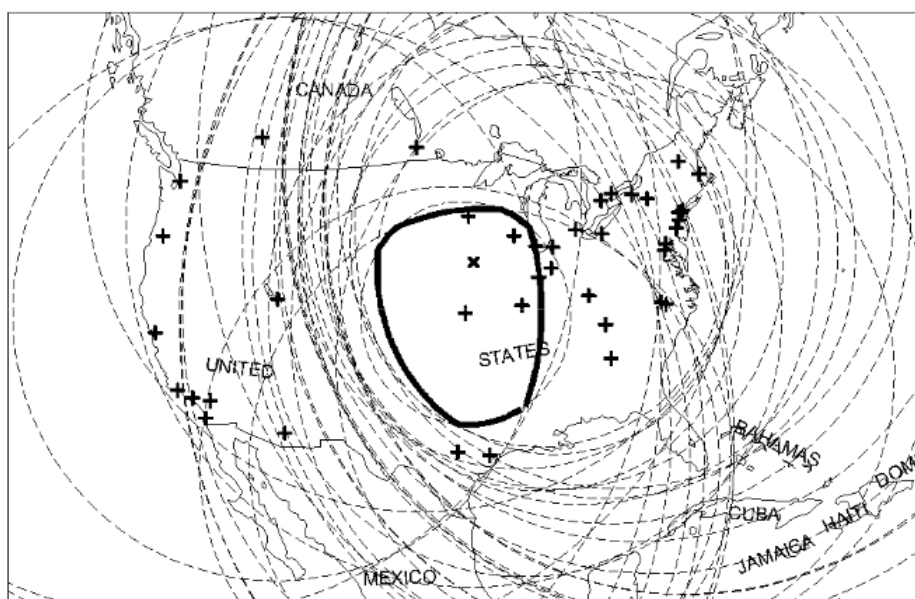
Aktívne metódy vyhľadávania geografickej polohy sieťového zariadenia pomocou dĺžky oneskorenia parametru RTT (Round-Trip Time) definuje čas ktorý paket prejde od zdroja k cieľu a naspäť. Toto oneskorenie môže byť ovplyvnené poklesom prenosovej rýchlosti, nepriaznivých vplyvov alebo geografickou vzdialenosťou. Parameter RTT využívajú funkcie na mapovanie oneskorovacích parametrov na geografickú vzdialenosť. Medzi aktívne geolokačné prístupy patria napríklad geoping alebo geolokácia založená na ohraničení (z angl. Constraint-Based Geolocation). Oba tieto prístupy využívajú výpočet založený na oneskorení paketu.

1.4.1 GeoPing

GeoPing určuje cieľ podľa mapovania orientačných bodov na lokalizáciu zariadenia. GeoPing predpokladá, že dve stanice ktoré sú blízko seba, budú mať rovnaké oneskorenie pri pingnutí okolitých orientačných bodov. Cieľový bod je otestovaný zo všetkých okolitých bodov a podľa týchto dát sa vypočíta vektor oneskorenia, ktorý slúži ako profil najbližších orientačných bodov. Cieľový bod je napokon určený z najpodobnejšieho profilu namapovaného bodu. Podobnosť sa vypočítava pomocou Euklidovskej metriky medzi dvoma vektormi. Táto vzdialenosť sa nazýva priestor oneskorenia (z angl. Delay space). [16]

1.4.2 Geolokácia založená na ohraničení

Namiesto mapovania zistovanej polohy, používa CBG (Constrain based geolocation) techniku podobnú triangulačnej technike kombinovania oneskorených časov z niekoľkých orientačných bodov a vracia pozíciu ktorá leží v priesečníku týchto bodov. Každý bod si meria oneskorenie ku ostatným známym bodom. Túto hodnotu si každý bod uloží a tým sa vytvorí priamka k funkcií oneskorenie k vzdialenosti. Predpoklad je že hľadaný bod sa nachádza v rámci vytvorenej kružnice, ktorej rádius je predpokladaná vzdialenosť. Tieto vzdialenosti sa následne skombinujú a ich priesečník predstavuje región v ktorom sa hľadaný bod nachádza.[14]



Obr. 1.1: Príklad metódy CBG prevzaté z [14]

2 DATABÁZY

Geolokačné služby sa pohybujú v cenovom rozmedzí od voľne dostupných cez služby, ktoré stoja stovky dolárov, až po služby v cenách desaťtisíce dolárov za mesiac. Okrem spomínaného DNS riešenia geolokačných služieb a proprietárnych API na dotazovanie polohy užívateľa (napr. Google Gears), existujú projekty ako HostIP.info, čo je voľne šíriteľný projekt, ktorý zbiera dáta od užívateľov ktorí sú jeho súčasťou. Okrem prispievania IP adries od poskytovateľov internetových služieb sú v projektoch ako tento zapojené aj webové stránky, v ktorých sa po návšteve užívateľa aktualizuje jeho fyzická poloha, a z toho sa následne stáva záznam databázy. Iný typ geolokácie vychádza z univerzít a vedeckých inštitúcií, kde často využívajú rôzne merania založené na metodológiách, ktoré sa snažia vylepšiť presnosť geolokačných služieb.[13]

2.1 Voľne dostupné databázy

V tejto práci sa na analýzu geolokačných dát využívajú ako komerčné tak aj nekomerčné databázy. Ako prvé sú popísané nekomerčné databázy. Pre prístup k týmto rozhraniam využívajú protokol HTTP a jeho metódy GET a POST. Všetky požadované informácie sú získavané vo formáte JSON (JavaScriptový objektový zápis - JavaScript Object Notation). Bližšia funkcionálna tejto aplikácie je spomenutá v kapitole 3.

2.1.1 freegeoip.net

Poskytovateľ freegeoip ponúka webové API na vyhľadávanie v ich geolokačnej databáze. Tento poskytovateľ dovoľuje 10 000 dotazov za hodinu. Táto databáza vracia informácie ako krajinu, región, mesto, PSČ, časovú zónu, zemepisnú dĺžku a šírku. Webový server freegeoip je voľne dostupný verejnosti¹

2.1.2 MaxMind

Spoločnosť MaxMind patrí medzi pionierov geolokačných služieb (založená v r.2002) a distribuuje rozsiahlu ponuku databáz od úrovne krajiny až po úroveň mesta. MaxMind ponúka svoje databázy ako voľne poskytovanú GeoLite2, ktorá je oproti platennej databáze GeoIP2 menej presnejšia. GeoLite2 ponúka dve možné databázy, a

¹<http://freegeoip.net/>

to GeoLite2Country a GeoLite2City. V tejto práci je použitá GeoLite2City kvôli presnejšej polohe na úrovni mesta.^[4]²

2.1.3 IP2Location

IP2location poskytuje na výber niekoľko voľne dostupných databáz, ktoré sa líšia množstvom požadovaných dát. V tejto práci je využitá databáza DB11lite. Táto databáza poskytuje informácie o krajine, regióne, meste, PSČ, časovej zóne a zemepisnej šírke a dĺžke. Lite verzia tejto databázy je ako v iných prípadoch taktiež obmedzená o počet záznamov a presnosť dát. Popis fungovania tejto databázy sa dá zhrnúť v nasledujúcich krokoch. Ako prvé sa získa IP adresa pomocou webového servera, následne sa IP adresa preloží na decimálny formát aby sa urýchlilo dotazovanie databázy. Ako posledné sa spätným vyhľadávaním IP čísla v IPLocation databáze určí geografická lokácia s danou presnosťou. Táto databáza je spravovaná spoločnosťou Hexasoft a okrem informácií o geografickej polohe ponúka aj informácie o počasi a šírke pásma.³^[7]

2.1.4 HostIP

HostIP sa líši od spomínaných databáz tým, že ide o open-source projekt podporovaný komunitou, ktorý spočíva v tom, že užívatelia ale aj niektorí poskytovatelia internetových služieb kontribuuju blokmi IP adries. Databáza má veľa blokov adries, z ktorých je približne 15.9% pridelených IP adries. Väčšina (63%) týchto blokov sa nachádza v severoamerickom regióne.⁴

2.1.5 DB-IP(IP to City)

Táto databáza obsahuje vyše 13 miliónov IPv4 a IPv6 blokov a označuje sa ako za jednu z najspoľahlivejších databáz čo sa týka presnosti. Najväčšie zastúpenie záznamov majú v tejto databáze Spojené štáty americké. Aj v tomto prípade tento poskytovateľ ponúka licencovanú, ale aj voľne dostupnú databázu. IP to City ponúka prístup k 9 499 952 IP záznamom databázy.⁵ ^[5]

²<https://www.maxmind.com/en/home>

³<http://www.ip2location.com/>

⁴<http://www.hostip.info/about.html>

⁵<https://db-ip.com/about/>

2.2 Komerčné databázy

V nasledujúcej sekcii sú spomenuté využité komerčné databázy v tomto projekte. Využité sú okrem spomínaných licencovaných verzií nekomerčných databáz aj komerčné databázy Skyhook, Geobytes a Neustar, ktoré umožňujú obmedzený počet dotazov za hodinu bez akýchkoľvek poplatkov. Tieto databázy sa označujú za veľmi presné s pomerne rozsiahlymi blokmi adries.[2]

2.2.1 DB-IP(IP to Location)

IP to Location je platený prístup k databáze DB-IP. Poskytuje všetky požadované informácie o polohe zariadenia spolu s poštovým smerovacím číslom, identifikátorom v databáze GeoNames a informácie o časovej zóne. Obsahuje 12 616 784 záznamov. [5]

2.2.2 Geobytes

Vyhľadávač IP Address Locator od spoločnosti geobytes ponúka vyhľadávanie 16 384 za hodinu, bez toho aby bolo potrebná komerčná licencia, čo je pre túto prácu postačujúce na jej otestovanie. Databázové rozhranie podporuje formát JSON. Priemerná dĺžka vyhľadávania v databáze je 4.5 sekúnd na vyhľadanie jedného záznamu. Je to z toho dôvodu aby bolo využité celé 60 minútové okno rovnomerne. Pomocou databázy Geobytes je možné získavanie množstva údajov. Okrem klasickým geolokačných informácií sa taktiež dá zistiť populácia, národnosť, hlavné mesto na úrovni krajiny a informácie o peňažnej mene krajiny.[10]

2.2.3 IP Info

Oproti databáze Geobytes ponúka IP info 1000 dotazov za deň. Vyhľadávanie prebieha ale rýchlejšie a to 10ms na jeden dotaz. IP bloky sú taktiež pravidelne aktualizované rôznymi poskytovateľmi internetových služieb. API poskytuje okrem lokácie aj informácie o organizácii, ktorej patri adresa a telefónna predvoľba krajiny.

2.2.4 MaxMind

Okrem nekomerčnej verzie databázy GeoIP je v tejto práci využitá aj databáza GeoIP Precision. Databáza je aktualizovaná každý týždeň novými adresovými blokmi a ponúka základne informácie o geografickej lokácii. Pri tejto platenej verzií sa jedná o presnosť záznamov 99.8% na úrovni krajiny, 90% na úrovni regionu, 87% na úrovni mesta v 50 kilometrovom okruhu.[4]

2.2.5 Skyhook

Spoločnosť Skyhook patrí medzi prominentné spoločnosti v oblasti geolokácie, ktorá prvotne začala v roku 2003 s geolokáciou prístupových bodov. Momentálne táto spoločnosť využíva GPS, Cell tower a senzory na širokej škále zariadení. Skyhook ponúka svoju proprietárnu službu Hyperlocal IP, ktorá využíva hybridné zistenie optimálnej lokácie pomocou Wifi, GPS a IP, telefónnych zariadení s údajnou presnosťou až do 10-20 metrov[8]

2.2.6 Eureka API

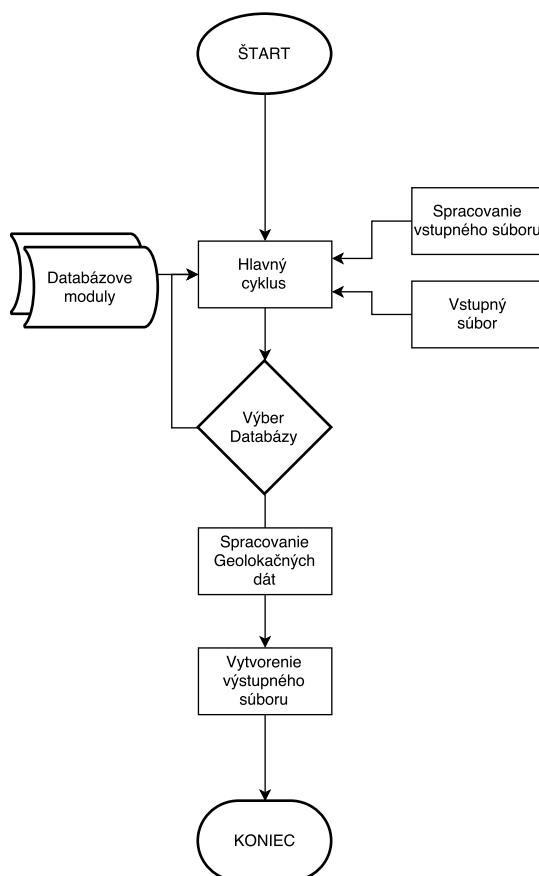
Toto API je vytvárané a udržiavané spoločnosťou IP-GeoLoc. Eureka podporuje formát JSON a poskytuje všetky základné geolokačné informácie, ktoré požadujeme. Táto databáza obsahuje cez 4 294 967 296 IPv4 adres a spracovanie dotazu je obdobne ako u Skyhook veľmi rýchle. [9]

3 NÁVRH APLIKÁCIE

Pri implementácii tejto aplikácie boli podľa pokynu vedúceho použité isté časti kódu z práce Ing. Štefana Zimu. Návrh tejto aplikácie spočíva v troch základných bodoch a to:

- Spracovanie vstupných dát.
- Porovnávanie dát s dátami geolokačných databáz.
- Analýza výstupných dát.

Pre vývoj tejto aplikácie bol zvolený jazyk Python, a to z dôvodu už existujúcich modulov, ktoré sú v tejto práci použité. Python, ako interpretovaný skriptovací jazyk spĺňa požiadavky pre tento projekt tak ako platformovú nezávislosť. Python je natívne podporovaný systémami na báze unixu. Syntakticky je Python veľmi prehľadný, ľahko čitateľný a obsahuje súhrne množstvo knižníc. Taktiež má Python široké využitie vo vývoji aplikácii pracujúcich na spodnej sieťovej vrstve.



Obr. 3.1: Diagram aplikácie.

Aplikácia využíva modulárnu štruktúru, ktorá bola zvolená z dôvodu nezávislosti jednotlivých modulov, prípadného rozšírenia funkcionality a celkovej prehľadnosti. Ako rozhranie pre použitie týchto modulov slúži modul `main.py`. V tomto module sa po overení verzie interpreta používanej užívateľom ako prvé naimportujú ostatne moduly ako `fileprocessing.py`, ktorý slúži na spracovanie vstupného súboru. Tento modul obsahuje funkciu ktorá, prechádza riadkami vstupného súboru a následne vracia list IP adries.

Po naimportovaní vlastných a vstavaných modulov sa pomocou funkcií modulu `argparse` spracujú príkazy zadané užívateľom do príkazového riadku. Aplikácia je postavená tak, aby si užívateľ mohol pri spustení vybrať ktoré z databáz uložených v slovníkovej štruktúre chce testovať. Užívateľ taktiež volí oddeľovač, ktorý reprezentuje ako sú vstupne a výstupné dáta oddelené, alebo či majú byť vykonané úpravy odrezanie a nahradenie. Následne sa pristúpi do funkcie `main` v module `main.py`. V tejto funkcii sa zavolajú databázy spolu s ďalšími požiadavkami zadanými užívateľom. Skript je možné spustiť s týmito prepínačmi:

- `-h` - pre spustenie skriptu s nápovedou.
- `-d` - pre výber databázy.
- `-i` - výber oddeľovača vstupného súboru.
- `-o` - výber oddeľovača výstupného súboru.
- `-v` - zvýšená verbálnosť počas behu programu.
- `-c` - odrezanie slov v súbore.
- `-r` - náhrada slov v danom súbore.
- `-m` - pre generovanie grafického výstupu.

3.1 Parsovanie vstupných dát

Príklad vstupného súboru používa túto dátovú štruktúru so záznamami v nasledujúcom poradí.

- IP Adresa.
- DNS.
- Kontinent.
- Krajina.
- Mesto.
- Zemepisná šírka.
- Zemepisná dĺžka.

Táto dátová štruktúra je parsovaná v súbore `fileprocessing.py` v ktorom je štruktúra definovaná, a to vo funkcii `getiprecords()`, ktorej návratová hodnota je táto

kolekcia. Tieto údaje môžu byť oddelené tabulátorom alebo bodkočiarkou. Pre oddelenie elementov tejto kolekcie reprezentujúcej vstupné dáta je použitý modul CSV (Comma separated values – hodnoty oddelené čiarkami), vďaka ktorému je možné vyčítať na základe oddelovača (bodkočiarka, medzera a tabulátor), že sa jedna o dáta ktoré sa následne uložia do kolekcie.

Najväčšiu časť tejto časti projektu predstavujú jednotlivé implementácie spomínaných databáz. Každá táto databáza je umiestnená vo vlastnom module. Každý z týchto databázových modulov obsahuje funkciu `check_ips`. Táto funkcia je odlišná pre každý modul, keďže niektoré moduly využívajú pre komunikáciu s databázou metódu HTTP GET alebo pomocou využitia webových elementov stránky a metódy POST na zadávanie adresy.

Všetky implementácie nekomerčných databáz využívajú metódu GET pre komunikácie. Dotazovaná je IP adresa v URL adrese. Funkcia `check_ips` prijíma nasledujúce parametre: ip adresu, separátor, voľba úpravy výstupného súboru a parameter `verbose`, ktorý vracia detailný popis priebehu funkcie.

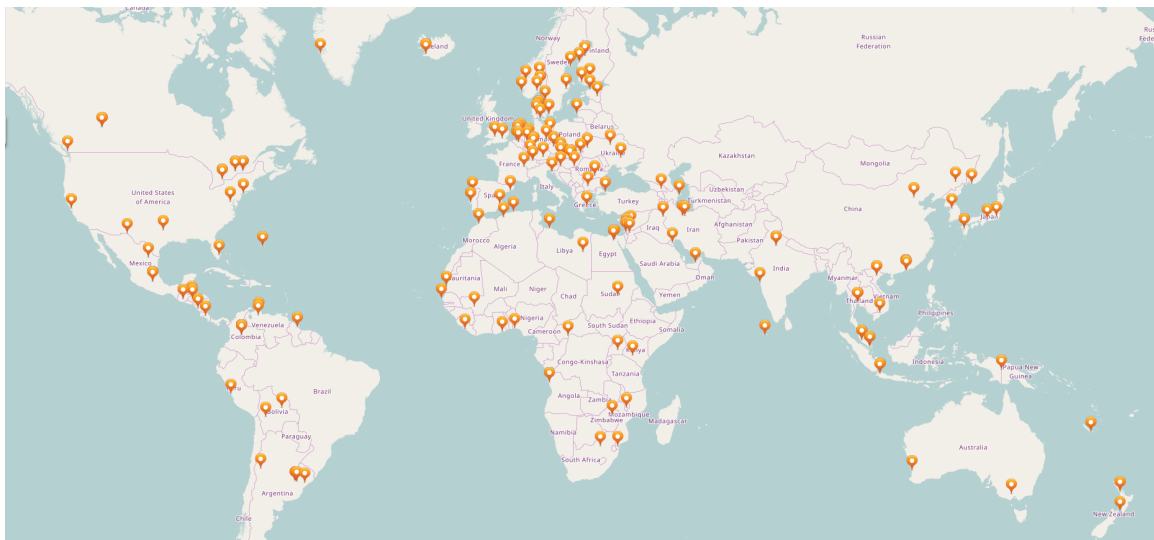
3.1.1 Príklad vstupných dát

Množina vstupných dát je v tomto projekte reprezentovaná textovým súborom, ktorý obsahuje záznamy v nasledujúcom formáte.

```
1 recordID ipAddress dnsName continent countryName regionName city
    unknownParameter unknownParameter latitude longitude
    dnsCorrection
```

Výpis kódu 3.1: Príklad štruktúry vstupných dát

Pre túto prácu boli zvolené dve veľké množiny záznamov, ktoré sa skladajú zo sieťových zariadení z rôznych oblastí sveta. Prvou testovanou množinou v tejto práci je množina o veľkosti 187 DNS serverov zo všetkých kontinentov Zeme. Väčšina týchto serverov sa nachádza najmä u internetových poskytovateľov a rôznych akademických inštitúciách. Táto množina bola zvolená z toho dôvodu, že geografická poloha týchto zariadení je takmer v každom regióne sveta do veľkej miery známa. Pre nájdenie polohy zariadení v tejto množine bola využitá databáza RIPE NCC (Réseaux IP Européens Network Coordination Centre). RIPE NCC využíva databázu WHOIS, ktorá slúži k evidencii údajov o majiteľoch internetových domén a IP adries. Pomocou tejto databázy bolo možné dosiahnuť takmer presnú lokáciu sieťových zariadení v testovanej množine DNS serverov na úrovni mesta a regiónu.[15]



Obr. 3.2: Rozloženie testovaných DNS serverov po kontinentoch.

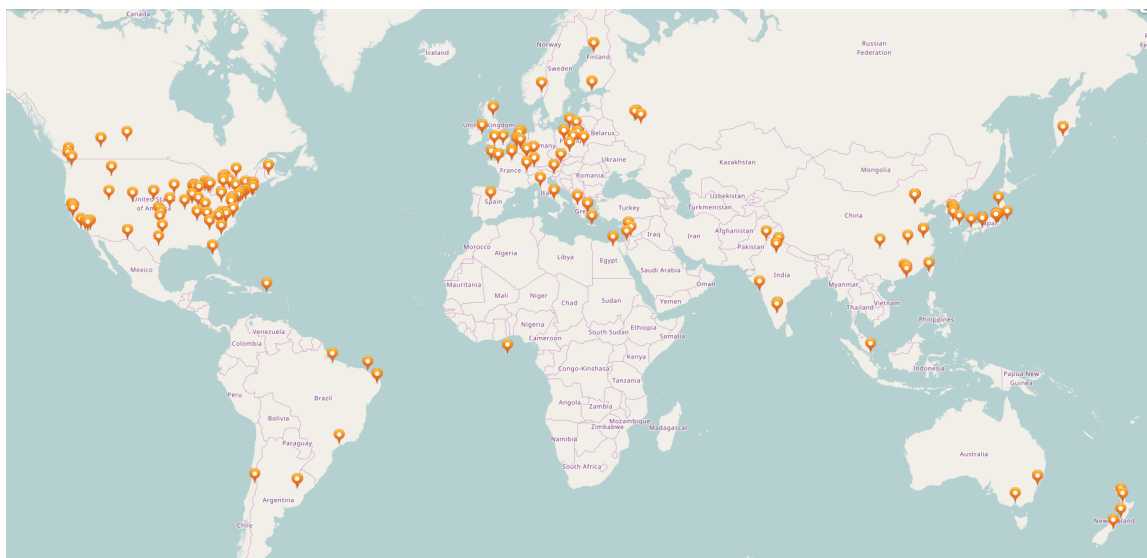
RIPE NCC je medzinárodná nezávislá organizácia, ktorá podporuje infraštruktúru internetu prostredníctvom technickej koordinácie v jej podporovaných regiónoch. Najdôležitejšou činnosťou je však fungovať ako Regionálny internetový register (Regional Internet Registry, podriadený inštitúcií IANA) poskytujúci globálne internetové zdroje a s nimi spojené služby (IPv4 a IPv6) pre členov v regióne RIPE NCC. Členovia sú najčastejšie poskytovatelia internetových služieb a rôzne akademické inštitúcie ako bolo spomenuté.[20]

Prvá veľká množina sú servery siete planetlab, ktoré slúžia pre výskum distribuovaných systémov, sieťových aplikácií a podobne. Uzly tejto siete sú rozložené po celej planete a v súčasnej dobe ich je vyše 1000. Pre účely tejto práce bolo testovaných 307 uzlov. Jedná sa predovšetkým o rôzne typy serverov. Sieť PlanetLab slúži najmä ako testovacie prostredie pre prekrývajúce sa siete. Prekrývajúce siete sú siete postavené na iných existujúcich sieťach. Vývojári pracujúci so sieťou PlanetLab môžu pre svoje účely pristupovať k množine zariadení, ktoré si rezervujú. V súčasnej dobe je v sieti PlanetLab vytvorených viac ako 600 projektov[17] a to napríklad:

- Zdieľanie súborov a vzdialené úložiská.
- Siete pre doručovanie obsahu (CDN – Content delivery Networks).
- Systémy na detekciu sieťových anomálií.
- Škálovateľné lokalizačné systémy.
- Nástroje na meranie činnosti siete.

Sieť PlanetLab má v porovnaní s inými experimentálnymi vývojovými sieťami, ktoré sú situované v menšej oblasti ako napríklad univerzitná testovacia sieť, tú výhodu,

že je možné vykonávať experimenty v reálnych podmienkach na oveľa väčšej mierke (čo poskytuje veľká množina geograficky rozptýlených serverov siete PlanetLab).[17]



Obr. 3.3: Rozloženie testovaných PlanetLab serverov po kontinentoch

Tab. 3.1: Príklad vstupných dát pre množinu DNS.

ID	IP Address	dns	Continent	Country	Region	city	latitude	longitude
10001	130.180.32.59	cf59413b.dynamic.tele-ag.de	EU	DE	Koeln	Koeln	50.93793	6.8744
10002	88.198.53.55	open-root.cesidian.info	EU	DE	Bavaria	Nuremberg	51.32	12.33
10003	185.93.180.131	frankfurt-ns01.cyberghostvpn.com	EU	DE	Hessen	Frankfurt am Main	44.43	26.1
10004	217.91.16.41	pd95b1029.dip0.t-ipconnect.de	EU	DE	Hessen	Darmstadt	49.87	8.58
10005	89.221.2.171	p59DD02AB.biz.antilo.de	EU	DE	Hamburg	Hamburg	51.49	7.42
10006	93.104.245.241	host-93-104-245-241.customer.m-online.net	EU	DE	Bavaria	Munich	48.1	11.47
10007	141.16.33.174	gate.bb-media-wlan-studioG.fta-berlin.de	EU	DE	Berlin	Berlin	52.5	13.28
10008	89.235.31.8	unknown	EU	CZ	South Moravia	Brno	49.09	16.27
10009	78.102.9.76	78.102.9.76.static.b2b.upcbusiness.cz.	EU	CZ	Bohemia	Prague	50.0651	14.433

Tabuľka 3.1.1 slúži príklad vstupných dat množiny DNS serverov. Každý záznam reprezentuje známu polohu zariadenia s údajmi o krajine, regióne, meste a zemepisnej šírke a dĺžke. K týmto údajom sa počas behu skriptu pridávajú hodnoty odhadnuté geolokacnou databázou, a následne sa vypočíta chyba pre každý riadok.

3.1.2 Získavanie geolokačných dát

V tejto sekcii je podrobnejšie riešený príklad implementácie modulu pre získavanie geolokačných dát prostredníctvom pasívnej metódy. Ako príklad je uvedený modul využívajúci databázu spoločnosti HostIP. Ako bolo spomenuté, každý modul na komunikáciu s databázou obsahuje metódu `check_ips`

```
1 def check_ips(ipRecords, separator, cut, replace, verbose):
2
3     for ipRecord in ipRecords:
4
5         request = request.get('http://api.hostip.info/' +
6                               + 'get_html.php?position=true&ip='
7                               + urllib.quote(ipRecord))
8
9         content = request.content
10        content_json = json.load(content)
11
12
13        if content_json.get('data'):
14            if content_json['data']['civic'].get('countryIso'):
15                country = content_json['country_code']
16
17            if content_json['data']['civic'].get('city'):
18                city = content_json['city']
19
20        if content_json['data']['location']:
21            latitude = content_json['latitude']
22            longitude = content_json['longitude']
```

Výpis kódu 3.2: Definícia funkcie pre získavanie dát z geolokačných databáz.

Ako prvé sa v každom takomto module vytvorí výstupný súbor, nastaví sa všetky premenné na predvolenú hodnotu a určí sa zvolený oddeľovač. Oddeľovač reprezentuje spôsob oddelenia v súbore vstupných dát. Následne sa nad kolekciou s IP adresami vstupného súboru začne vykonávať cyklus, ktorý je ukončený až po dosiahnutí konca riadku vstupného súboru. Pre využitie metódy GET je použitý modul `urllib(urllib.request)`. Tento modul definuje funkcie a triedy pomocou ktorých je možné pristupovať k URL. Vytvorí sa request objekt na ktorom sa zavolá jeho metóda `get` z URL databázy a IP adresou. Návratová hodnota tejto funkcie sa otestuje porovnaním hlavičky protokolu HTTP či obsahuje kód 200, čo reprezentuje kladný stav. V prípade že je tento kód iný ako táto hodnota je vyvolaná výnimka čo znamená

že, API vrátilo prázdnu hodnotu alebo návratová hodnota z servera bola negatívna (stavový kód 4xx – chyba na strane klienta alebo 5xx – chyba na strane serveru). Celý obsah novo získaných dát sa uloží do premennej `content`, ktorý zabalí dáta do JSON objektu. Formát JSON takto umožňuje veľmi jednoduché pristupovanie k jednotlivým položkám pre bližšie spracovanie.

```
1  # ERROR CALC
2      error = '-'
3      if latitude != '-' and longitude != '-':
4          correct = (float(ipRecord.latitudeCoordinate), float(
5                      ipRecord.longitudeCoordinate))
6          retrieved = (float(latitude), float(longitude))
          error = vincenty(correct, retrieved).kilometers
```

Výpis kódu 3.3: Príklad výpočtu chyby.

Ďalším krokom je výpočet chyby databázy. Pri výpočte sa berie do úvahy chyba v zemepisnej dĺžke a šírke. Získané hodnoty z databázy sa porovnávajú s tými vo vstupnom súbore. Informácie o zemepisnej dĺžke a šírke v vstupnom súbore boli získané pomocou GoogleMaps API podľa geografickej polohy inštitúcie, v ktorej by sa zariadenia mali nachádzať. Na výpočet chyby sa využíva Vincentyho vzorec. Jedná sa o výpočet využívajúci iteratívnu metódu používanú v geodetike na výpočet dvoch bodov vzdialených od seba na povrchu sféroidu. Pre tento výpočet sa používajú dve metódy. Prvá priama metóda vypočítava pozíciu jedného bodu a následne azimut z druhého bodu. Druhá inverzná metóda počíta geografickú vzdialenosť a azimut medzi dvoma danými bodmi. Tieto metódy sú považované za veľmi presne a to až s presnosťou 0.5mm na zemskom elipsoide. Pre tento výpočet je použitá knižnica `geopy`¹, ktorá obsahuje implementáciu tohto výpočtu, konkrétne druhej metódy. Knižnica `geopy` je použiteľná na lokáciu koordinátov adries, miest a krajín na zemi. `Geopy` využíva geokódery tretích strán a iné zdroje dát. `Geopy` obsahuje triedy geokóderov ako napríklad Google Geocoding API (V3)². Po výpočte chyby sa všetky získané dáta zapisujú do výstupného súboru.

¹<https://github.com/geopy/geopy>

²<https://developers.google.com/maps/documentation/geocoding/start>

```

1 latitude = str(latitude)
2     longitude = str(longitude)
3     error = str(error)
4
5     output_file.write(separator.join(ipRecord.row) + separator
6         + "dpip2city" + separator + country +
7             separator + country_match + separator +
8             region + separator +
9             region_match + separator + city + separator +
10                city_match +
11                separator + latitude + city + separator +
12                city_match +
13                separator + latitude + separator + longitude
14                + separator +
15                error + "\n")

```

Výpis kódu 3.4: Príklad zápisu do súboru.

Po zapísaní sa zatvoria všetky otvorené súbory dát sa skript ukončí. Skript sa ukončuje aj v prípade že užívateľ zadal zlý názov databázy.

API voľne prístupných databáz využíva REST (Representational state transfer) čo je v architektúra rozhrania, ktorá umožňuje pristupovať v dátam získaných API pomocou HTTP metód GET, POST, PUT a DELETE. Pre všetky implementácie modulov bola použitá metóda GET. Príklad požiadavku s metódou GET je možné vidieť vo vyššie uvedenej ukážke implementácie. Dotaz s použitím metódy GET má návratovú hodnotu vo formáte JSON. Tento odľahčený formát patrí v dnešnej dobe medzi štandardne zaužívaný v rôznych webových aplikáciach a službách, hlavne kvôli svojej čitateľnosti a zapisovateľnosti pre bežného človeka. Je založený na podmnožine jazyka JavaScript a je popísaný štandardom ECMA-262³ Slúži na výmenu dát.

JSON je založený na dvoch štruktúrach:

- Kolekcia párov názov/hodnota. Tá je označovaná v mnohých jazykoch ako objekt.
- Zoradený zoznam hodnôt.

Na prácu s týmto formátom JSON je použitá knižnica JSON⁴, táto knižnica ponúka jednoduchý spôsob parsovania JSON-u a mapovania na objekty v Pythone. Príkladom využitia tejto konverzie je využitie metódy `json.loads()` ktorá namaňuje JSON do kolekcie slovníka, čo uľahčuje ďalšiu prácu s týmito dátami.

³<https://www.ecma-international.org/publications/standards/Ecma-262.htm>

⁴<https://docs.python.org/3.5/library/json.html>

V nasledujúcej ukážke implementácie je možné vidieť aj využitie metódy GET na získanie týchto dát.

```
1 request = request.get('http://api.hostip.info/get_html.php?
   position=true&ip=' + urllib.quote(ipRecord))
2 content = request.content
3 content_json = json.load(content)
```

Výpis kódu 3.5: Príklad vytvárania objektu request.

Hodnota, ktorá sa načíta do premennej request je nasledovná:

```
1 {
2   "address": "144.160.98.31",
3   "country": "US",
4   "stateprov": "California",
5   "city": "Los Angeles",
6   "latitude": "33.9802",
7   "longitude": "-118.44"
8 }
```

Výpis kódu 3.6: Obsah objektu request.

Takúto odpoveď vracia každá z implementovaných databáz. Niektoré z testovaných databáz posielajú okrem týchto dát aj dodatočné informácie ako napríklad poštové smerovacie číslo a časovú zónu.

3.1.3 Pravidlá pre úpravu dát

Z práce Ing. Štefana Zimu bolo taktiež použité jeho implementácia metódy na úpravu dát pomocou vytvárania pravidiel. Princíp týchto pravidiel spočíva v odrezávaní a nahradzovaní reťazcov v poliach obsahujúcich údaje z databáz. Vytváranie pravidiel prebieha na základe analýzy výstupného súboru a následného vytvorenia pravidiel.

Na analýzu slúži skript `analyzer.py`. Analýza súboru je spúšťaná pomocou uvedenia argumentu `-p` pri spustení programu. Samotná analýza prebieha na základe prechádzania jednotlivých poli v zázname. Každý modul obsahuje polia pre zhodu a nezuhodu na úrovni krajiny, regiónu alebo mesta. Potom, čo je nájdený reťazec, ktorý obsahuje hľadané slovo, je identifikovaná nezuhoda a pristupuje sa k samotnému vytváraniu pravidiel. Pravidlo môže byť buď nahradzovanie (replace) alebo

odrezávanie (cut) získaného parametru. Pri pravidle **replace** sa prechádzajú záznamy a v prípade nesprávne uvedeného záznamu v niektorom z polí je nahradená novým reťazcom. V prípade operácie **cut** sa taktiež jedná o iteráciu celého súboru a po nájdení požadovaného reťazca je tento reťazec odstránený.[18]

3.2 Implementácia spracovania dát

Po spracovaní jednotlivých výsledkov požadovaných dotazov do výstupných súborov sú tieto súbory spracované pomocou knižnice **pandas**. **Pandas** pod licenciou BSD (Berkeley Software Distribution) a poskytuje využitie dátových štruktúr a nástrojov na analýzu dát a manipuláciu. Knižnica obsahuje dátové štruktúry ako dátový rámec (**DataFrame** – Objekt pre manipuláciu dát s integrovaným indexovaním) alebo **Series** (Jednorozmerná dátová štruktúra obsahujúca heterogénne dáta). V tejto práci boli použité práve tieto dve štruktúry pre bližšie spracovanie a vyhodnotenie výsledkov. Výsledné súbory sa v module **fileprocessing.py** načítavajú do dátového rámca, pričom každému výstupnému súboru pripadá jeden dátový rámec.

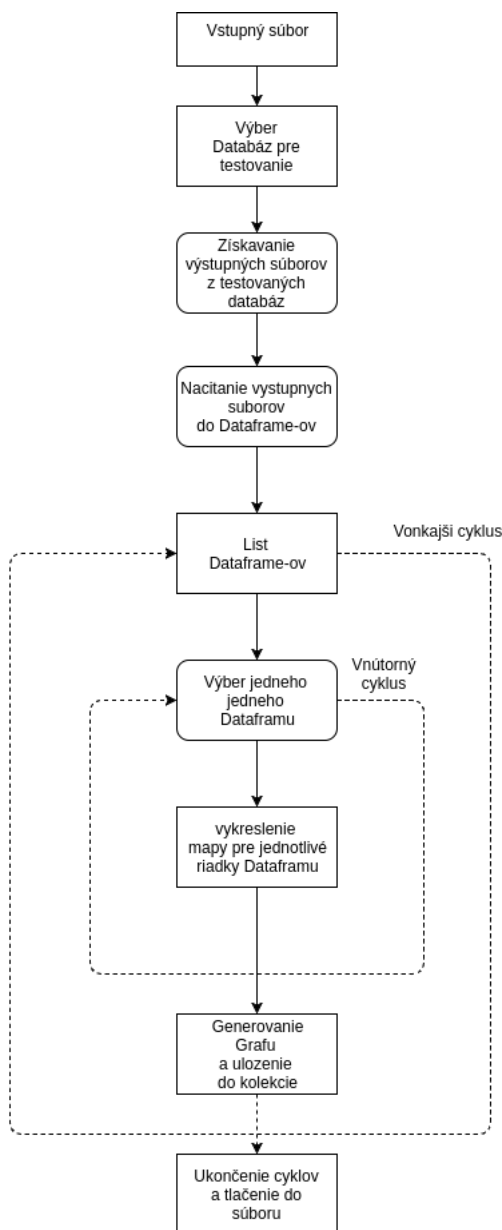
Pomocou tejto štruktúry je možné jednoducho vykonávať rôzne operácie na zozbieraných dátach. Program v tejto práci generuje Medián a Kvantil pre chybu jednotlivých zozbieraných dátových množín. Pre vizuálne spracovanie výsledných dát je použitá knižnica **folium**⁵. **Folium** je knižnica napísaná v jazyku Python pre zobrazenie máp pomocou JavaScript knižnice **leaflet.js**⁶. Pomocou tejto knižnice je možné zobrazovať manipulované dáta v jazyku Python ako geografické body na interaktívnej mape.

V nasledujúcom diagrame je podrobnejšie opísaná funkcia modulu **fileprocessing.py**, ktorý slúži na spracovanie výstupných súborov. Tento modul obsahuje dve funkcie ktoré sú volané v module **main.py**. funkcia **get_ip_records** slúži na spracovanie IP adries vstupného súboru na kolekciu list. Tento list je následne využitý v jednotlivých moduloch, ktoré dotazujú testované databázy. Druhou hlavnou funkciou v tomto module je funkcia **process_output** na spracovanie výstupného súboru ktorej funkčnosť je znázornená v diagrame. Táto funkcia má za úlohu spracovať výstupné dáta do jednotlivých dátových rámcov. Pre každú databázu, teda výstupný súbor sa vytvorí jeden dátový rámec ktorý sa uloží do listu. Následne je použitá sekvencie dvoch vnorených cyklov. Vonkajší cyklus ktorý prechádza jednotlivé dátové rámce, a v každom kroku tohto cyklu sa vykoná vnútorný cyklus. V každom takomto cykle sa prechádzajú jednotlivé riadky výstupného súboru a extrahujú sa z nich výsledné dáta ako: geografické súradnice, vypočítaná

⁵<https://github.com/python-visualization/folium>

⁶<http://leafletjs.com/>

Vincentyho chyba, informácia o zhodách na úrovni mesta, regiónu a krajiny.



Obr. 3.4: Diagram modulu `fileprocessing.py`.

Na nasledujúcich stranach je znázornená funkčnosť aplikácie. Ako prvé je zadaný vyber databázy s požadovanými prepínačmi, zvoleným vstupným súborom `input.dat` a oddelovačom použitým pre vstupný a výstupný súbor. Záznam z vstupného súboru sa načíta do skriptu ktorý je následne spracovaný. Po ukončení skriptu, je vygenerovaný HTML súboru, ktorý obsahuje mapu s bodmi ktoré sa nachádzajú na geografickej pozícií (obr. 3.5). Pre každý riadok výstupného súboru sa vytvárajú dva body, a to bod pôvodnej pozície, ktorá bola zadaná vo vstupnom súbore a bod odhadovanej pozície, ktorú vracia geolokačná databáza.

```

1 /usr/bin/python3.5 "/Ziskavanie Geolokacnych Dat/main.py" -d
   Freegeoip -i tab -v -o tab input.dat
2 Input file to be used: input.dat
3 Databases to be used: Freegeoip
4 Input file record delimiter: tab
5 Output file record delimiter: tab
6 File used for cutting:
7 File used for replacement:
8 Correct input record identified by 'correct=1' last field in
   _IP_RECORD tuple. Incorrect record identified by 'correct=0'
   field.
9 _IP_RECORD(id='050000', ip='129.240.228.138', dns='planetlab2.
   simula.no', continent='EU', countryCoordinate='NO',
   regionCoordinate='Oslo', cityCoordinate='Oslo',
   unknown_parameter_1='http://www.simula.no', unknown_parameter_2
   ='Simula Research Laboratory', latitudeCoordinate='59.93',
   longitudeCoordinate='10.75', dns_correction='diffIpFromDns', row
   =['050000', '129.240.228.138', 'planetlab2.simula.no', 'EU', 'NO
   ', 'Oslo', 'Oslo', 'http://www.simula.no', 'Simula Research
   Laboratory', '59.93', '10.75', 'diffIpFromDns'], correct=1)

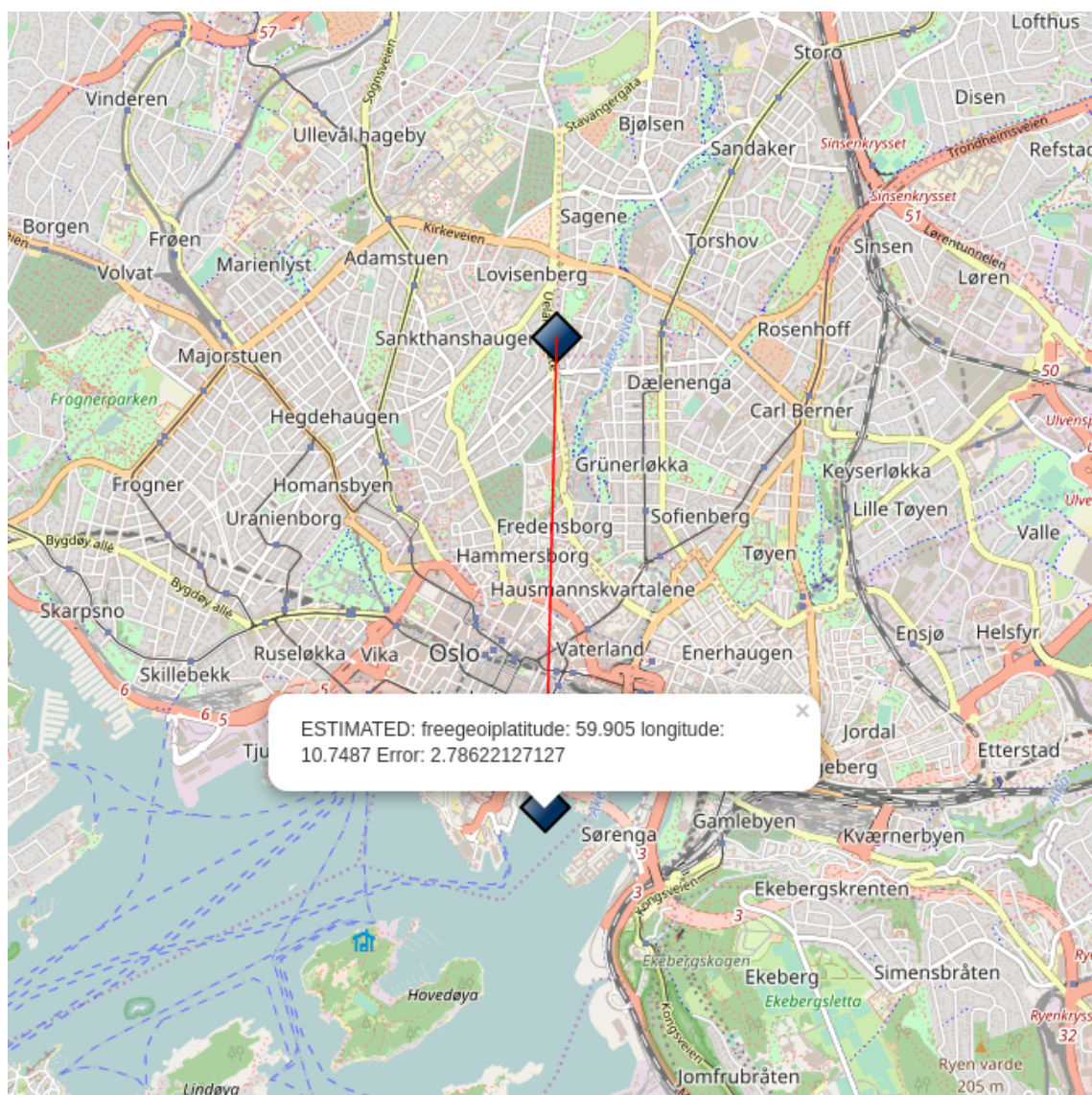
```

Výpis kódu 3.7: Príklad spustenia skriptu.

Jedným z výstupov programu je taktiež grafické zobrazenie dát na ktoré je použitá knižnica matplotlib. Matplotlib je knižnica pre jazyk python určená pre 2D zobrazenie číselných hodnôt. V práci je knižnica matplotlib⁷ využitá na zobrazenie kumulatívnej distribučnej funkcie, pre hodnoty chyby vincentyho dĺžky pre všetky databázy. Tieto grafy sú generované v module `fileprocessing.py`.^[19]

Aplikácia dokáže testovať viac databázy naraz, a výsledky pre každý záznam vstupného súboru spracováva do súboru html s interaktívnou mapou. Na obr. 3.6 je vidieť príklad takéhoto súboru. Do mapy sa zakresľujú vždy dva koordinaty pre jednu databázu, a to pôvodnú pozíciu a pozíciu odhadnutú databázou.

⁷<https://matplotlib.org/>



Obr. 3.5: Príklad mapy generovanej skriptom pre jeden záznam



Obr. 3.6: Príklad mapy generovanej skriptom pre jeden záznam.

4 ANALÝZA VÝSTUPNÝCH DÁT

Posledná kapitola tejto práce pojednáva o analýze získaných dát. Vzhľadom na limity stanovené niektorými geolokačnými databázami sú testovacie vzorky obmedzené na počet 187 testovaných DNS serverov a 307 staníc siete PlanetLab. Z týchto dvoch vzoriek je ale možné určiť presnosti jednotlivých databáz a objektívne ich porovnať. Pri testovaní však vznikli isté nedostatky a to najmä pri databáze DB-IP, ktorá vracia slovný názov krajiny miesto dvoj písmenového kódu, ktoré vracajú iné databázy.

Tab. 4.1: Výsledky testovanie pre množinu DNS.

Databázy	Vstupné záznamy (187)								
	Krajina			Región			Mesto		
	Zhoda	Nezhoda	Neznáme	Zhoda	Nezhoda	Neznáme	Zhoda	Nezhoda	Neznáme
DB-IP	0	187	0	0	187	0	0	187	0
EurekAPI	174	13	0	42	103	42	70	80	37
Geobytes	136	50	0	34	152	0	61	125	0
IPInfo	175	12	0	49	113	25	94	65	28
MaxMind	175	12	0	0	306	94	167	142	91
Skyhook	142	7	38	22	60	105	30	27	130
freegeoip	106	11	70	47	140	0	106	81	0
Geolite2city	177	11	0	0	123	64	110	78	0
IP2Location	0	187	0	49	138	0	90	97	0

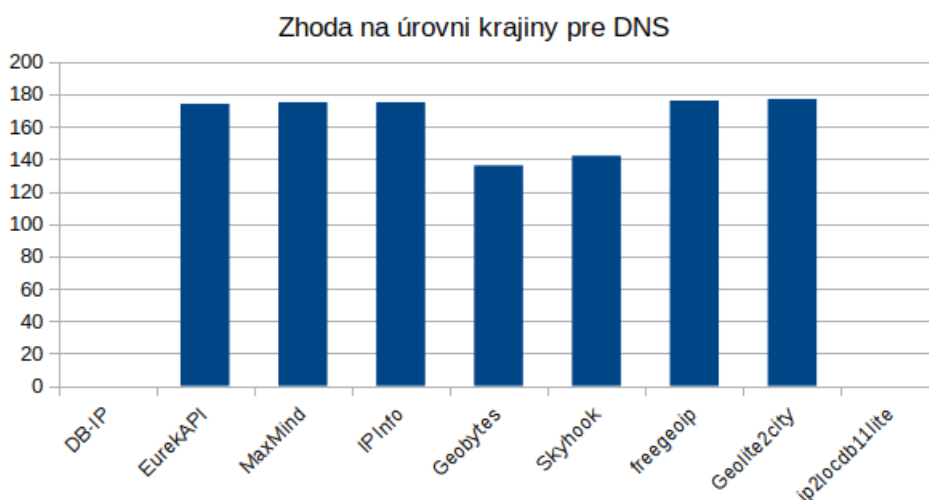
V tabuľkách je možné vidieť počet zhody na jednotlivých úrovniach, pričom najväčší počet zhody je na úrovni krajiny, kde sa pri testovaní vzorky DNS serverov nepodarilo správne určiť geografickú pozíciu len trom databázam. Najväčšie nezhody nastávajú na úrovni regiónu, kde prevláda veľmi malý počet pre zhodu. Môže to byť spôsobené najmä jazykovými rozdielmi pre jednotlivé regióny a samozrejme nedostatočne presnými informáciami databáz o regiónoch. Zhoda na úrovni mesta je na tom vo výraznejšej miere lepšie ako pri regiónoch čo sa týka správneho odhadu zariadení a o niečo menším počtom neznámych staníc.

4.1 Výsledky testovania množiny DNS serverov

4.1.1 Zhoda na úrovni krajiny

Jednotlivé databázy možno čo najobjektívnejšie hodnotiť podľa odhadu krajiny, kde sa dá hovoriť o veľkej úspešnosti pre takmer všetky databázy. Poloha krajiny je pre geolokačné databázy najjednoduchším získateľným parametrom. Veľké nedostatky v rozsahu záznamov je vidieť najmä u databázy DB-IP, pri ktorej možno hovoriť o

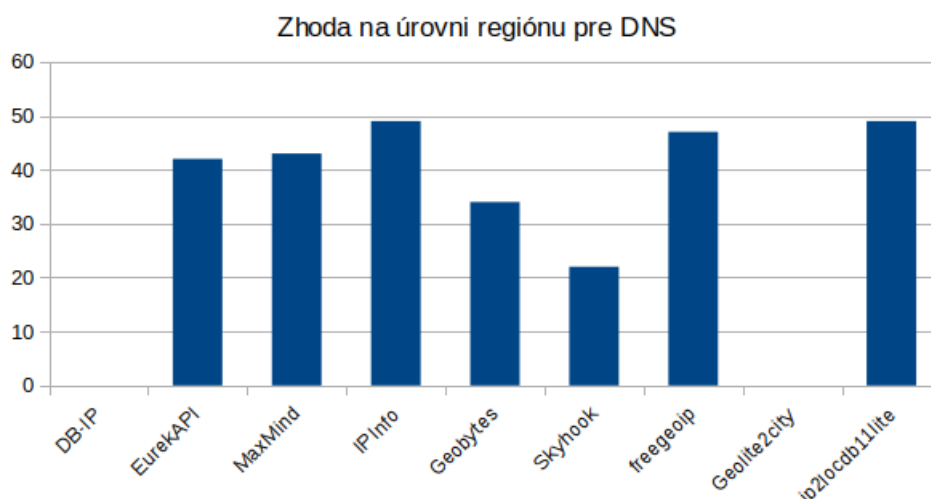
100% nezhode pre túto testovaciu množinu. To iste je možné vidieť aj pri databáze IP2Location. Najväčšiu presnosť pri zhode na tejto úrovni je vidieť u databázy Geolite2city. Jedna sa o verziu tejto databázy ktorá obsahuje najväčší počet záznamov zo všetkých ponúkaných verzií databáz od spoločnosti MaxMind. Táto databáza dosahuje počet zhôd až 177 z 187 testovaných zariadení. Komerčnej databáze freegeoip sa podarilo správne odhadnúť 176 zariadení na tejto úrovni. Druhá testovaná databáza od spoločnosti MaxMind je databáza GeoIP2 ktorej sa podarilo správne odhadnúť 175 z 187 zariadení. Identický výsledok dosahuje aj databáza IPInfo, ktorej sa podarilo odhadnúť rovnaký počet zariadení. Databázy Eureka sa taktiež podarilo odhadnúť porovnateľné množstvo zariadení ako v predošlých databázach. Následne za týmito databázami nasledujú Skyhook, Geobytes. Výsledky zobrazuje obrázok.



Obr. 4.1: Výsledky zhody na úrovni krajiny pre DNS.

4.1.2 Zhoda na úrovni regiónu

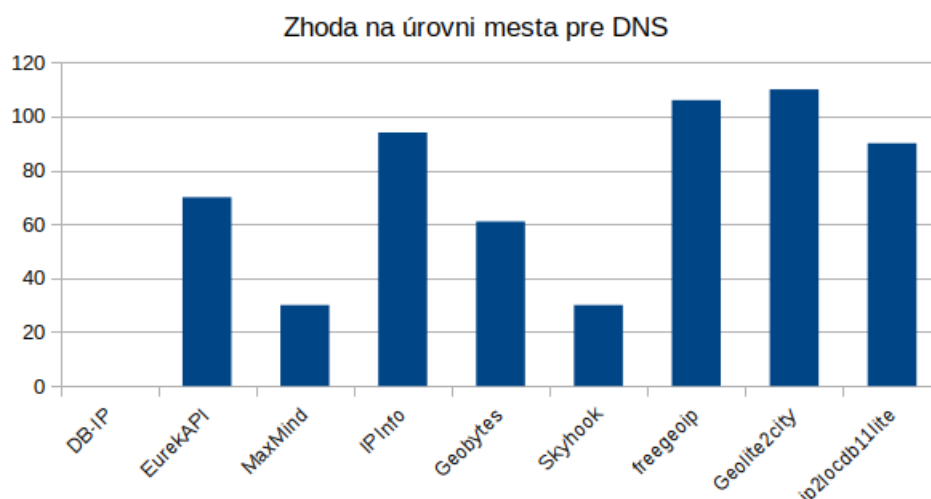
Presnosť na úrovni regiónu je pre komerčné aj nekomerčné databázy problematická. Je to hlavne z dôvodu rozdielného jazykového zápisu v záznamoch jednotlivých databáz. Program bol implementovaný tak, aby bolo možné čo najbližšie identifikovať región. Z výsledkov je zjavné že ani tento implementačný detail nebol dostačujúci a možno teda usúdiť, že testované databázy nepoužívajú jednotný štandard pre označovanie regiónov. Väčšine databáz s výnimkou DB-IP a db11lite sa podarilo odhadnúť podobný počet zariadení.



Obr. 4.2: Výsledky zhody na úrovni regiónu pre DNS.

4.1.3 Zhoda na úrovni mesta

Vzhľadom na to, že databázy obsahujú takmer rovnaké pomenovania miest, je na tejto úrovni počet správnych odhadov o niečo lepší. Pri niektorých databázach je návratová hodnota tohto parametru v inom jazyku ako v anglickom, čo spôsobuje odlišnosti pri vyhodnotení výsledkov. Najväčší počet zhody na tejto úrovni zaznamenala databáza od spoločnosti MaxMind, a databáza freegeoip. Najmenšie hodnoty zhody dosahovala databáza DB-IP. Najväčší počet chýbajúcich záznamov bolo pri databáze Skyhook.



Obr. 4.3: Výsledky zhody na úrovni mesta pre DNS.

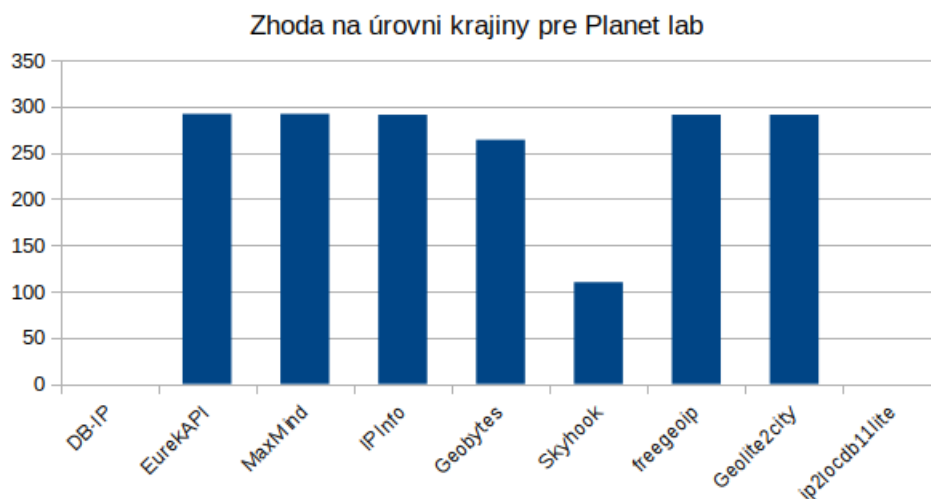
4.2 Výsledky testovania množiny PlanetLab.

Tab. 4.2: Výsledky testovanie pre množinu PlanetLab.

Databázy	Vstupné záznamy (307)								
	Krajina			Región			Mesto		
	Zhoda	Nezhoda	Neznáme	Zhoda	Nezhoda	Neznáme	Zhoda	Nezhoda	Neznáme
DB-IP	0	201	106	0	0	307	0	0	307
EurekAPI	292	15	0	3	273	31	131	146	29
Geobytes	264	43	0	2	305	0	110	197	0
IPInfo	291	16	0	0	261	46	131	133	43
MaxMind	292	15	0	0	285	22	132	149	26
Skyhook	110	2	195	0	67	240	30	31	246
freegeoip	291	16	0	0	307	0	134	173	0
Geolite2city	291	16	0	0	305	2	133	174	0
IP2Location	0	307	0	1	306	0	118	189	0

4.2.1 Zhoda na úrovni krajiny

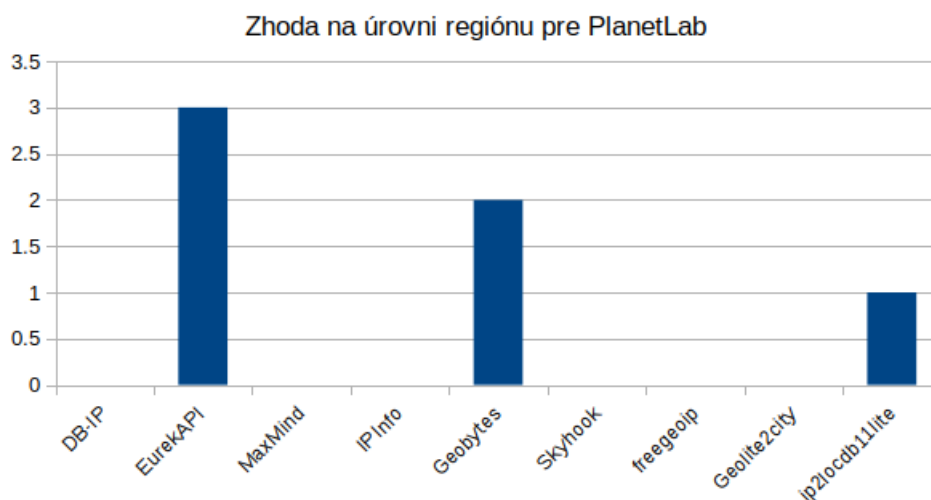
Výsledky testovania množiny planetLab sa na úrovni krajiny význačne nelíšia od výsledkov testovania množiny DNS serverov. Aj v prípade tohto testovania sú nedostatky v databáze Skyhook, ktorá má najväčší počet chýbajúcich údajov z testovaných databáz na tejto úrovni.



Obr. 4.4: Výsledky zhody na úrovni krajiny pre PlanetLab.

4.2.2 Zhoda na úrovni regiónu

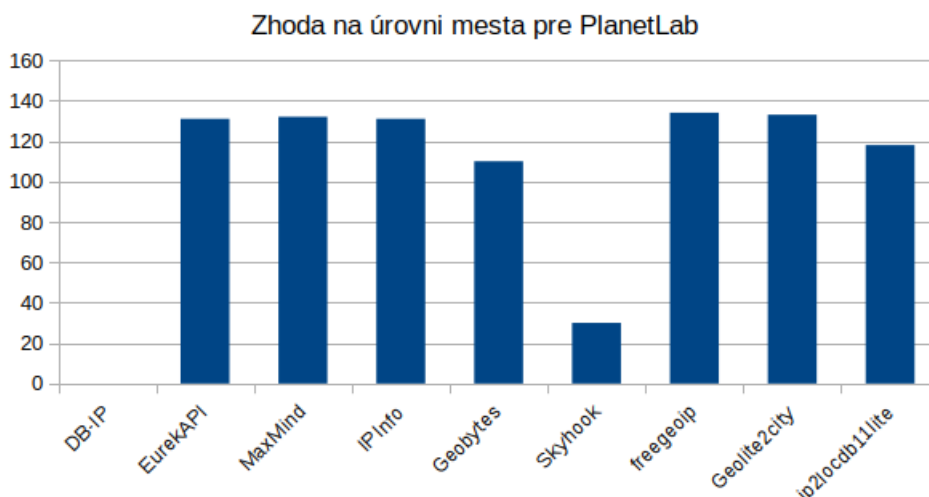
Aj v tomto prípade boli výsledky testovania vysoko nepresné z dôvodu zle odhadnutých regiónov databázami. V bližšom porovnaní s testom pre množinu DNS serverov je možné vidieť oveľa väčšie nedostatky pri tomto testovaní. Pri porovnaní výstupných súborov databáz je vidieť, že väčšina databáz tento parameter nedokázala správne určiť. Tento nedostatok mohol vzniknúť z toho dôvodu, že väčšina testovaných zariadení tejto vzorky sú virtuálne stroje, pri ktorých je ťažšie určiť presnú pozíciu podľa IP adresy.



Obr. 4.5: Výsledky zhody na úrovni regiónu pre PlanetLab.

4.2.3 Zhoda na úrovni mesta

V testovaní na úrovni mesta je vidieť iné výsledky oproti testovaniu predošlej množiny a to hlavne v náraste správnych odhadov pre databázy Maxmind GeoIP2 a Eureka. Zvyšné testované databázy dosiahli pomerne podobné výsledkov. Databáza Skyhook dosiahla na tejto úrovni najväčší počet neznámych odhadov a podarilo sa jej správne odhadnúť 30 záznamov. Po aplikovaní pravidiel pre databázu DB-IP dokázala táto databáza správne odhadnúť až 142 záznamov. Pri porovnaní s testovaním množiny DNS je vidieť, že testy pre obe množiny na tejto úrovni dokázali správne odhadnúť takmer 50% z testovanej vzorky záznamov. S výnimkou databázy Skyhook mali takmer všetky databázy nulový nezhodný odhad, okrem databáz IPInfo, MaxMind GeoIP2 a Eureka. Z tohto poznatku je možné vyvodiť že tieto databázy obsahujú rozsiahle záznamy IP adres.



Obr. 4.6: Výsledky zhody na úrovni mesta pre PlanetLab.

4.3 Porovnanie dosiahnutých výsledkov

Pri testovaní pomocou zavedení pravidiel pre výmenu celoslovných názvov krajín a miest sa výsledky databázy DB-IP značne zlepšili v prípade oboch testovaných množín. Táto databáza dosahovala vysoké výsledky v zhode na úrovni krajiny a mesta porovnateľne s výsledkami databáz MaxMind GeoIP2 a IPInfo.

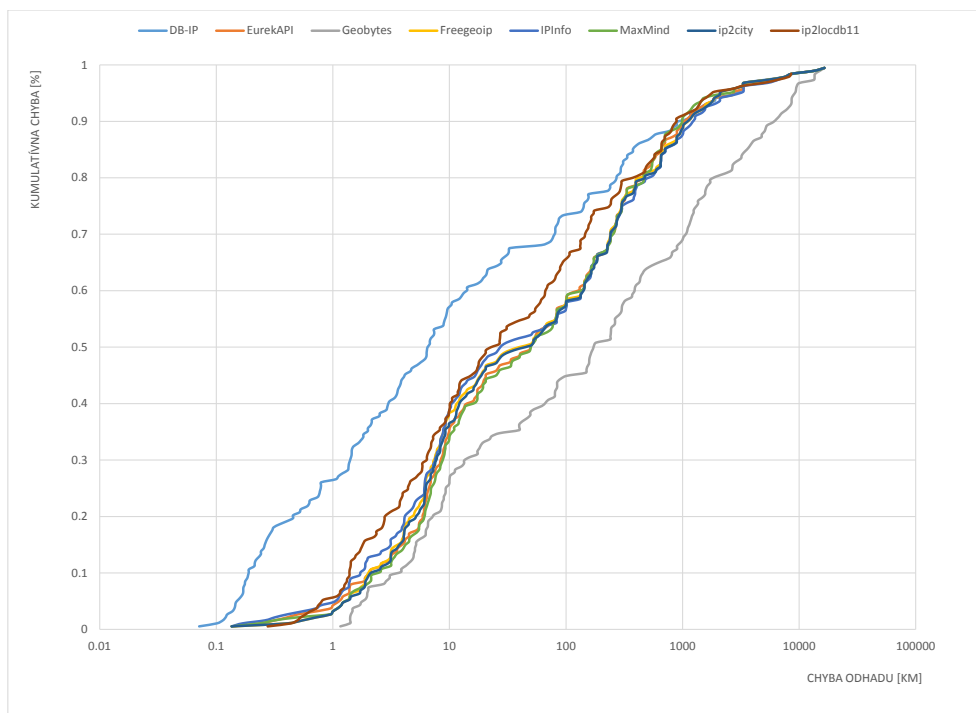
Pre obe testované množiny bola taktiež aplikáciou vypočítaná kumulatívna distribučná funkcia, ktorá je zobrazená v grafoch 4.7 a 4.8, Táto funkcia je počítaná z chyby vypočítanej Vincentyho dĺžkou pri porovnávaní pôvodných koordinátou záznamu s tými ktoré boli získané testovanými databázami. Na krivkách je zobrazená presnosť jednotlivých databáz. Čím vyššie sa nachádza na osi kumulatívnej chyby tým je vyššia pravdepodobnosť chyby pri odhade. V tabuľkách 4.3 a 4.4 je možné vidieť hodnoty mediánu a kvantilov pre všetky tieto databázy.

V celkovom hodnotení testovaných databáz zo získaných výsledkov pre tieto množiny, boli najkladnejšie vyhodnotené komerčné databázy Eureka, MaxMind, IPInfo, databázu Geobytes je podľa získaných výsledkov hodnotená ako priemerná. Databáza DB-IP obsahuje vo výslednej analýze pre obe množiny najslabšie výsledky. Je to z toho dôvodu že táto databáza využíva odlišnú konvenciu názvov pre odhadované parametre (krajina, región, Mesto). Avšak po aplikovaní pravidiel pre optimalizáciu použitých z práce Ing. Štefana Zimu sa výsledky tejto databázy javia na úrovni presnosti databáz Eureka, MaxMind a IPInfo. Svedčia o tom aj výsledky na obr. 4.8 a 4.7 kde je vidieť že databáza DB-IP dosahovala podobné výsledky pri určovaní zemepisnej šírky a dĺžky ako kladne hodnotené databázy. Z nekomerčných

Tab. 4.3: Kvartily chýb jednotlivých databáz pre množinu DNS serverov.

Databáza	Q_0	$Q_{0.25}$	$Q_{0.5}$	$Q_{0.75}$	Q_1
dbIpToLoc	0.071	0.792	6.776	142.539	14517.282
skyhook	0.235	6.753	34.830	292.887	16517.302
maxMindGeoIp2Pre	0.135	7.277	51.646	302.046	16454.104
freegeoip	0.136	6.163	49.638	299.712	16517.715
ipInfo	0.135	6.141	29.192	310.213	16517.716
geobytes	1.159	9.922	177.468	1330.1970	16518.009
eurekAPI	0.135	6.818	50.407	301.513	16517.716
ip2city	0.135	6.360	51.074	300.979	16517.716
IP2Location	0.275	4.511	26.683	237.550	16517.717

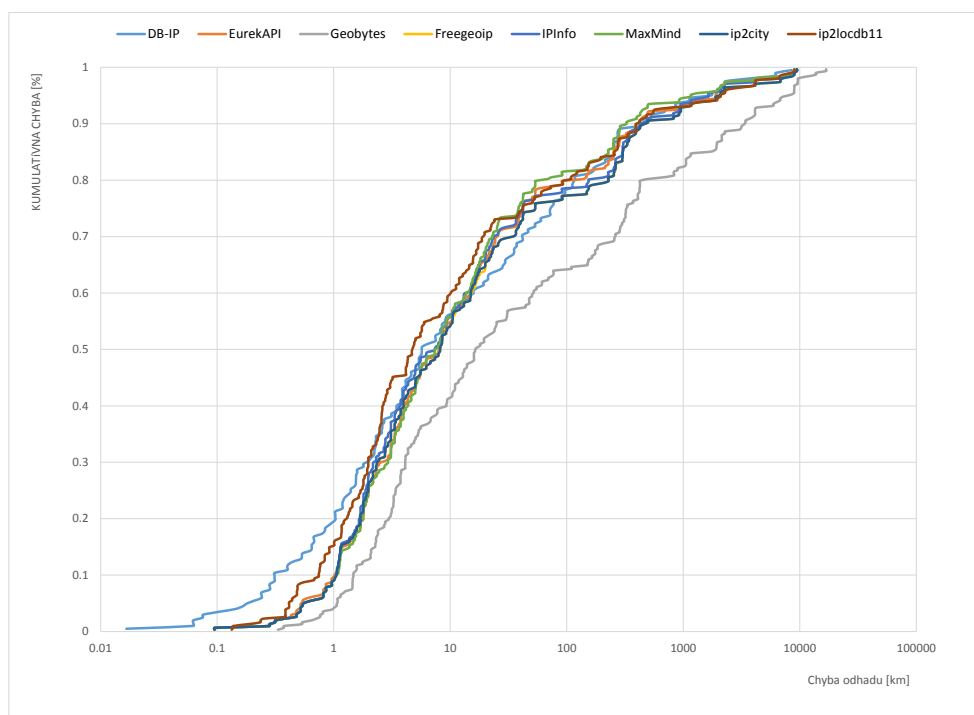
databáz dosahovali najpresnejšie výsledky databázy IP2Location (ip2locdb11lite), Geolite2city, a Freegeoip, ktorých výsledky boli na úrovni kladne hodnotených komerčných databáz. tieto dve databázy sú označené ako veľmi presne, čo potvrdzujú aj obr. 4.8 a 4.7 z ktorých je vidieť že databázy sú porovnateľné z komerčnými databázami pri odhade zemepisnej šírky a dĺžky. V práci nebolo možné testovať databázu HostIP kvôli trvajúcej nefunkčnosti API tejto databázy od marca 2017.



Obr. 4.7: Kumulatívna distribučná funkcia pre množinu DNS.

Tab. 4.4: Kvartily chýb jednotlivých databáz pre množinu PlanetLab.

Database	Q_0	$Q_{0.25}$	$Q_{0.5}$	$Q_{0.75}$	Q_1
dbIpToLoc	0.016	1.483	5.75	73.354	8544.765
skyhook	0.013	0.455	2.943	21.655	5170.66
maxMindGeoIp2Pre	0.094	1.912	7.463	42.037	9434.434
freegeoip	0.094	1.976	8.212	53.602	9434.436
ipInfo	0.094	1.912	7.463	42.037	9434.436
geobytes	0.331	3.426	16.906	326.642	16799.061
eurekAPI	0.094	1.986	8.081	41.938	9434.44
ip2city	0.094	1.966	8.208	53.602	9433.113
IP2Location	0.133	1.768	4.783	41.935	8973.979



Obr. 4.8: Kumulatívna distribučná funkcia pre množinu PlanetLab.

5 ZÁVER

Táto práca sa zaoberá analýzou komerčných a voľne dostupných databáz. Všetky z testovaných databáz poskytujú geografické informácie na troch úrovniach. Táto práca sa zaoberá analýzou presnosti na úrovni krajiny, regiónu a mesta. Následne je na dvoch testovaných vzorkách vyhodnotená presnosť týchto parametrov. V práci boli použité dve testovacie vzorky. Tá prvá je množina DNS serverov z všetkých kontinentov sveta, a tá druhá bola vzorka uzlov experimentálnej siete PlanetLab.

V úvode tejto práce boli riešené teoretické poznatky o IP geolokácií a rôznych technikách získania geografických dát pomocou IP adresy. V tejto časti je taktiež zahrnutý rozbor jednotlivých testovaných databáz.

Jedným z cieľov tejto práce bolo navrhnúť a implementovať aplikáciu na zbieranie dát z geolokačných databáz. Táto aplikácia následne vyhodnocuje a graficky spracováva do podoby interaktívnej mapy v ktorej je možné vidieť a porovnať pôvodnú pozíciu testovaných zariadení a ich pozíciu odhadovanú geolokačnými databázami. Aplikácia taktiež generuje graf kumulatívnej distribučnej funkcie na základe chyby odhadu v kilometroch a tabuľku z vypočítanými hodnotami kvantilov. Implementačné detaily sú predmetom druhej časti tejto práce. V rámci tejto sekcie sú bližšie riešené spomenuté funkcionality aplikácie. Aplikácia je písaná v jazyku python a to z dôvodu jednoduchosti tohto jazyka pri spracovaní dát pomocou knižníc ako pandas, ktorá bola v tejto práci použitá na spracovanie výsledných dát, a knižníc na vizualizáciu ako matplotlib a folium.

V záverečnej časti tejto práce je analýza výsledkov testovania geolokačných databáz. Testovaných bolo 187 zariadení pre množinu DNS serverov a 307 zariadení pre množinu zariadení PlanetLab. Z testovaných databáz boli najlepšie výsledky dosiahnuté databázami IPInfo, Geolite2City od spoločnosti MaxMind, databáza Eurek, IP2location a databáza freegeoip. Veľmi presné výsledky boli dosiahnuté databázou DB-IP a to najmä na úrovni mesta a krajiny. Pri tejto databáze vznikala nepresnosť, a to kvôli inej mennej konvencii parametrov ako pri iných databázach.

LITERATÚRA

- [1] PUŽMANOVÁ, Rita. *TCP/IP v kostce. 2., upr. a rozš. vyd.* České Budějovice: Kopp, 2009. ISBN 978-80-7232-388-3.
- [2] Huffgaker, B. and Fomenkov, M. and claffy, k. *Geocompare: a comparison of public and commercial geolocation databases - Technical Report*, Cooperative Association for Internet Data Analysis (CAIDA), May 2011,
- [3] Yuval Shavitt and Senior Member and Noa Zilberman, *A geolocation databases study*, IEEE Journal on Selected Areas in Communications [cit.29.5.2017]. Dostupne z URL: <<http://www.eng.tau.ac.il/~shavitt/pub/JSAC2011-Geolocation.pdf>>
- [4] MaxMind *IP Geolocation and Online Fraud Prevention* [online]. [cit. 10.5.2017]. Dostupne z URL:<<https://www.maxmind.com/en/home>>
- [5] DB-IP *IP Geolocation and Network Intelligence* [online]. [cit. 10.5.2017]. Dostupne z URL: <<https://db-ip.com/>>
- [6] *IP Address Geolocation to Identify Website Visitor's Geographical Location* [online]. [cit.10.5.2017]. Dostupne z URL: <<http://ipinfo.io/>>
- [7] *IP Address Geolocation to Identify Website Visitor's Geographical Location* [online]. [cit. 10.5.2017]. Dostupne z URL: <<https://www.ip2location.com/>>
- [8] SkyHook *SkyHook / Global Location and Context Software Products* [online]. [cit. 10.5.2017]. Dostupne z URL: <<http://www.skyhookwireless.com/>>
- [9] EurekaAPI *IP Address Geolocation to trace Country, Region, City, ZIP Code, etc* [online]. [cit. 10.5.2017]. Dostupne z URL: <<https://www.eurekapi.com/>>
- [10] Geobytes *Get City Details* [online]. [cit. 10.5.2017]. Dostupne z URL: <<http://geobytes.com/get-city-details-api/>>
- [11] Wang Jinxia, Xu Xiaoyan, Yang Min, Zang Tianning, *IP Geolocation Technology Research Based on Network Measurement*
- [12] Ingman Poesse and Steve Uhlig and Mohamed Ali Kaafar and Benoit Donnet and Bamba Gueye, *IP Geolocation Databases: Unreliable?*, ACM SIGCOMM Compute Communication Review, April 2011
- [13] Jayaprabha Bendale, Prof. J. Ratanaraj Kumar, *Review of Different IP Geolocation Methods and Concepts* International Journal of Computer Science and Information Technology, Vol. 5, 2014

- [14] *Proceedings of the 2006 ACM SIGCOMM Internet Measurement Conference: IMC 2006* : October 25-27, 2006, Rio de Janeiro [i.e. Janeiro], Brazil. New York: Association for Computing Machinery, c2006. ISBN 1-59593-561-4.
- [15] RFC3912, L. Daigle, WHOIS Protocol Specification, Internet Requests for Comments, RFC, 3912, september 2004,, 2070-1721, RFC Editor, [cit. 8.4.2017] Dostupne z URL <<https://tools.ietf.org/html/rfc3912>>
- [16] IONESCU Daniel, *Geolocation 101: How It Works, the Apps, and Your Privacy* pcworld, 2010. [cit.10.5.2016]. Dostupne z URL:<<http://www.pcworld.com/article/192803/geolo.html>>
- [17] PlanetLab: an open platform for developing, deploying, and accessing planetary-scale services. [online], 2007, [cit. 10.5.2016]. Dostupné z URL: <<https://www.planet-lab.org>>
- [18] ZIMA,Š.*Analýza geolokačních databází*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 65 s. Vedoucí bakalářské práce doc. Ing. Dan Komosný, Ph.D.
- [19] Hunter, J. D., Matplotlib: A 2D graphics environment, Computing In Science & Engineering, [online], 2007, [cit. 17.5.2016]. Dostupne z URL: <<https://matplotlib.org/index.html>>
- [20] "About US." *RIPE Network Coordination Centre* e. N.p., n.d. Web. 17 May 2017. Dostupne z URL <<https://www.ripe.net/about-us>>

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

URL	Jednotná adresa zdroje – Uniform Resource Locator
API	Programovateľné aplikačné rozhranie – Application Programming Interface
GPS	Globálny polohový systém – Global Positioning System
GPS	Globálny polohový systém – Global Positioning System
XML	Rozšírený značkovací jazyk – Extensive Markup Language
JSON	JavaScriptový objektový zápis – JavaScript Object Notation

ZOZNAM PRÍLOH

A Obsah priloženého CD

50

A OBSAH PRILOŽENÉHO CD

Priložené CD obsahuje elektronickú verziu tejto práce, zdrojové kódy projektu, databázové súbory databáz a knižnice pre prácu s API databáz. Táto práca je taktiež na servere github kde je udržiavána a dostupná k stiahnutiu ([Link na github](#)) .

```
/
├── resources
│   ├── helpers
│   ├── nonCommerciallDBs (Implementácie modulov nekomerčných databáz)
│   ├── commerciallDBs (Implementácia modulov komerčných databáz)
│   ├── databaseFiles (Databázové súbory)
│   ├── dependencies (Knižnice pre prácu s API databázami)
│   └── fileprocessing.py (Modul pre spracovanie dat)
├── results (Súbor obsahujúci výstupné súbory)
├── main.py (Spustací skript aplikácie)
├── input.dat (Súbor vstupných dat)
└── README.md
```